

N94-18338

1.1.1

3rd NASA Symposium on VLSI Design 1991

## Experience with Custom Processors in Space Flight Applications

M. E. Fraeman, J. R. Hayes, D. A. Lohr, B. W. Ballard,  
R. L. Williams, and R. M. Henshaw  
Johns Hopkins University Applied Physics Laboratory  
Laurel, Maryland 20723

**Abstract-** APL has developed a magnetometer instrument for a Swedish satellite named Freja with launch scheduled for August 1992 on a Chinese Long March rocket. The magnetometer controller utilized a custom microprocessor designed at APL with the Genesil silicon compiler. The processor evolved from our experience with an older bit-slice design and two prior single chip efforts. The architecture of our microprocessor greatly lowered software development costs because it was optimized to provide an interactive and extensible programming environment hosted by the target hardware. Radiation tolerance of the microprocessor was also tested and was adequate for Freja's mission—20 kRad(Si) total dose and very infrequent latch-up and single event upset events.

### 1 Introduction

The Johns Hopkins University Applied Physics Laboratory (APL) has developed a microprocessor that is well suited to one-of-a-kind embedded applications especially in satellite instrument control. The chip has been qualified for use in a magnetometer instrument for the Swedish Freja satellite. The processor's language directed architecture reduced Freja software costs because the flight hardware served as its own development system. Thus, unlike traditional interpreted programming languages like Basic, Lisp, or Smalltalk, our Forth language development system was fully supported on the embedded flight processor. Performance was also equivalent or better than that obtained by other microprocessors programmed in languages like C with traditional cross-compilers and development systems.

Our experiences using Forth to program spacecraft instrumentation computers, and our early efforts to design a 32-bit microprocessor specifically intended to execute Forth code are described in this paper. The design, architecture, and performance of our most recent version of this microprocessor, called the SC32<sup>1</sup>, are summarized in Section 4. Discussion of our use of the SC32 in the Freja magnetometer includes our efforts to qualify the microprocessor for space flight. Finally, we discuss some of the lessons we learned using a custom designed integrated circuit in space flight hardware.

---

<sup>1</sup>The SC32 has been commercially licensed by Silicon Composers, Inc., Palo Alto, Ca. They offer chips, board level development systems, and support software.

Table 1: APL Forth-based Subsystems and Experiments

SPACECRAFT	SUBSYSTEM/EXPERIMENT	LAUNCH DATE	PROCESSOR
MAGSAT	Attitude Control	6/79	RCA 1802
DMSP	Magnetometer	(classified)	RCA 1802
HILAT	Magnetometer	6/83	RCA 1802
Polar Bear	Magnetometer	6/83	RCA 1802
Astro-1	Ultraviolet Telescope(HUT)	12/90	AMD 2900
Freja	Magnetometer	8/92 ( <i>est</i> )	SC32

## 2 Background

### 2.1 Forth

Forth has an extremely simple syntax so only a trivial parser is needed to allow it to run in impoverished hardware environments. Lexical properties are also simple. Forth subroutines, called words, are delimited by spaces. The words themselves can consist of any characters other than the delimiter. This simplicity keeps the interpreter small allowing full featured Forth systems to fit comfortably in as little as 8 kbytes of memory.

Programming in Forth consists of defining new words in terms of existing words. The new word is incrementally compiled and can be invoked interactively by the programmer. Thus, the usual benefits of interpreted languages are reaped, especially simplified testing and a resulting higher confidence in program correctness.

### 2.2 APL Space Applications of Forth

Table 1 summarizes APL's experience with spacecraft instrumentation we have developed and programmed using Forth.[1] We have also used the language on other projects including ground support equipment and control of laboratory instrumentation. Application tasks ranged from relatively simple data acquisition functions to control of the complex, space shuttle based Hopkins Ultraviolet Telescope (HUT)—one of three ultraviolet telescopes (all programmed in Forth) that comprised the Astro-1 mission at the end of 1990. Our most recent instrument, a magnetometer for the Swedish Freja satellite will be described later in this paper.

Our earliest space flight applications were based on the relatively simple RCA 1802 microprocessor. But during the early definition of the HUT command and data handling system around 1980, it became clear that a far more powerful processor was needed to satisfy that project's requirements. After exploring an architecture based on as many as four TI 9900 microprocessors (the fastest microprocessor qualified for space that was then available), we realized that a single faster machine would have numerous advantages. The software would be easier to write and test, and more importantly, uniprocessor code and hardware would be more flexible in the face of evolving requirements and as system

interfaces were more clearly defined.

## 2.3 The Hopkins Ultraviolet Telescope Processor

The AMD 2900 bit-slice component family was used to build a 16-bit computer that implemented Forth's primitive operations directly in microcode. In the early 1980s, this was the only way we could build a single processor with throughput that met our requirements and that also could be qualified for use in space. Our bit-slice processor was able to compile and execute Forth interactively, even on the flight unit, without needing extensive support tools. Performance was also very good (approximately 500,000 Forth operations per second) which allowed us to design an unusually flexible software system. The final flight software required about 5 person-years of development time (including developing the detailed software requirements), contained 29 cooperating concurrent processes, and consisted of about 12,000 lines of Forth code and comments.

We gained valuable experience with Forth based computers while developing, using, and flying the HUT processor. A fast computer that supported a compact but interactive and extensible software development system on flight hardware had many advantages. It encouraged the development of powerful yet flexible software while minimizing the costs of writing, testing, and maintaining that code. However, HUT also showed that the 64 Kword address space of 16-bit machines was inadequate for larger embedded systems. Towards the end of the development cycle flight processor memory became too full to support an interactive environment so we had to fall back on clumsier traditional cross-compiler based methodology.

## 3 The FRISC Project

At the same time our work on HUT hardware was winding down in 1984, we were also initiating an effort to develop experience in VLSI design. We combined our experience in Forth computers and our interest in VLSI into an effort to develop a 32-bit Forth microprocessor. During 1985 we developed the processor architecture that we called FRISC (Forth Reduced Instruction Set Computer) and ported VLSI design tools developed at several universities a 68010 based workstation.

### 3.1 FRISC 1

By the beginning of 1986, with tools and architecture firmly in hand, we started detailed design of a chip that implemented most of our ideas. This was FRISC 1, the first in a series of chips that evolved into the SC32. We targeted the 4  $\mu\text{m}$  Silicon on Sapphire (SOS) process then available through MOSIS. We selected SOS technology for several reasons. First, SOS is inherently immune to radiation induced latch-up and would thus be a candidate technology for future integrated circuits used in flight systems. The absence of active-substrate junction capacitance reduces load and hence improves speed. Circuit density is improved because there is no minimum p-active—n-active separation design rule.

#### 1.1.4

Finally, on a more practical note, the SOS process was available through MOSIS at no cost as far as the project's budget was concerned. So the chance to get experience with a technology with significant benefits for chips intended for use in space was too good to pass up.

Design of the 18,000 transistor chip was completed by mid-April 1986. It easily fit inside a standard MOSIS 7.9 mm x 9.2 mm pad frame. We used *caesar* for layout, *lyra* for design rule checking, *rn1* for functional simulation, *spice* for circuit simulation, and the usual collection of customized shell scripts, format translators, and system utilities for coordinating the design team's work. While the chips were being fabricated, we built a wire wrapped Multibus CPU board with memory and a programmable non-overlapping clock generator board to test our parts.

Three months later we had our chips and began to test them. About half of the parts that were eventually delivered appeared to function except that one data bit was always stuck high. Unfortunately, that specific bit was used in the instruction set to cause the processor to output a value, so we had no way to inspect the contents of the chip's registers. Microscopic analysis later revealed a spacing design rule violation at the interface between the pad ring cell and the cell containing the chip's interior logic. This error was undetected because *lyra* flattened the layout of intersecting areas on adjacent cells after checking the cells individually. Our design hierarchy consisted of the pad ring in one cell and all the other circuitry in a second cell completely enclosed by the ring. Therefore the top level rule check flattened the entire design and greatly exceeded the maximum virtual memory space supported by our host workstation so our mistake went undetected.

Despite this layout error, one chip was fully functional and we were able to demonstrate a full Forth system running on our own custom 32-bit microprocessor. But before we could submit a corrected design, MOSIS announced that they would no longer offer access to SOS.

## 3.2 FRISC 2

At the beginning of 1987, we started to redesign our chip with the MOSIS scalable ( $3\text{ }\mu\text{m}$  to  $1.2\text{ }\mu\text{m}$ ) bulk CMOS process. We also used the *magic* layout editor instead of *caesar* but still depended on *rn1* for switch level simulation. By April we sent the layout to MOSIS for a 20,000 transistor chip that implemented almost all of our original architecture. The active area for this chip, designed with  $3\text{ }\mu\text{m}$  feature sizes, was slightly smaller than the previous version but it still required a 7.9 mm x 9.2 mm pad frame. However, an inadvertently grounded substrate prevented that part from working. Using a combination of infrared microphotography and careful inspection of the layout in the hot region we eventually located the error.<sup>2</sup> Since we made our mistake, a circuit extractor called *meztra*, was modified at the University of Washington to specifically detect similar errors. Apparently we weren't the first, and based on errors we've detected in other designs, not the last group to make a substrate connection error.

---

<sup>2</sup>This error has since been missed by dozens of students taking the midterm exam in a JHU VLSI design class.

A corrected layout was fabricated shortly thereafter and was fully functional. The fixed FRISC 2 could execute about 2.5 million Forth primitives per second (about five times faster than 25 MHz Motorola MC68020 running Forth) and consumed 150 mW. However, this performance was about twice as slow as we expected due to an incorrectly sized control line driver.

## 4 The SC32

While our efforts had eventually produced a functional and usable microprocessor, we did not reach our design goals on first silicon. In fact, we felt that our small team would not be able to build chips much more complex than FRISC 2 with the tools and workstations we used for that design. Furthermore, full logical and parametric functionality would probably be achieved only after several fabrication iterations. Our simulations were not as thorough as we would have liked since our workstation required a day to complete a switch level simulation of the execution of a few machine instructions. Determining the impact of more than one or two architectural alternatives on chip speed and area was impractical. Irregular structures such as control logic were very tedious to layout. Minor changes in control logic would often result in days of work to resimulate and update the layout. As our speed problem with FRISC 2 demonstrated, these structures were also a likely source of parametric as well as functional errors.

### 4.1 Genesil

Rather than waiting several years for workstation speeds to improve before tackling more complex chip designs, we investigated commercial VLSI design tools. Silicon Compilers Inc. (now part of Mentor Graphics, Inc.) had just released the Genesil silicon compiler. This was a fully integrated set of VLSI tools that let the user describe, implement, and analyze a design at the block diagram level.

Genesil's intended market was logic designers with no VLSI experience. Yet we were attracted to it because the compiler allowed a user to easily and quickly investigate the implications of many architectural alternatives. We felt that the greatest improvements in system performance could be gained by optimizing architecture while lower level enhancements would be of secondary importance. Any inefficiencies introduced by the high level design tool should be more than compensated for by the better architecture that the silicon compiler would allow the designer to develop. Genesil also automated many of the most time consuming aspects of VLSI design so a small team would be able to tackle larger projects. Thus we hoped that Genesil would be the better tool that would let our small team tackle larger designs.

### 4.2 SC32 Design

Genesil was installed at our site by June 1987, and we started using it to explore approaches to implementing our Forth architecture. We also enhanced our computer's architecture

based on the experience we gained on our earlier designs. The greater complexity that Genesil let us tackle with the same size team (2-3 part time people) also allowed us to improve the architecture. By mid-November we had completed our Genesil design work including thorough simulations of thousands of instructions. But due to delays in design verification at Silicon Compilers, our mask level design wasn't delivered to the foundry until February, 1988. After an extra month delay caused by problems with test vector formats, we received fully functional tested parts in May. The next day we had a single board computer running an interactive Forth development system.

We consider this third version of our Forth processor a complete success. It was fabricated with a 2  $\mu$ m epitaxial CMOS n-well process, contained 35,000 transistors, and consumed 660 mW. The die was 9.9 mm x 9.6 mm and was packaged in an 84 pin ceramic pin grid array. Despite obvious inefficiencies in the overall chip layout, the processor still ran at 10 MHz. Because the processor architecture is optimized for Forth the comparatively slow clock rate speed still executed 8-12 million primitives per second—a throughput still unmatched by any other 32-bit microprocessor implementation of the language of which we are aware.

### 4.3 Architecture

The detailed architecture of the SC32 has been described elsewhere.[2] Briefly, the machine has a 32 bit word address architecture and an instruction set that can implement most Forth primitives in a single instruction. Flow control instructions specify an absolute destination address and execute in a single cycle with no delay slots. The machine's register set is organized into two top-of-stack caches with single cycle access within the instruction set to the top four locations of each stack. These on-chip caches support stack depths limited only by main memory with overflow and underflow events handled entirely by hardware. Less than 1% overhead is added to typical Forth programs by our approach to stack management. There are up to eight other utility and special purpose registers allowed. The data path allows arithmetic operations between these registers to be completed in a single cycle. A flexible load/store instruction format transfers data between registers and memory and can also be used to form literal values.

### 4.4 Performance

Measuring and comparing processor performance is always controversial—especially for a new architecture not supported by commonly used languages. Different implementations of Forth are also difficult to compare since there are no commonly used benchmark programs written in that language. Finally, it is only natural to ask how a Forth version of a program compares to an equivalent implementation in a more widely used language.

Since Forth is the only high level language available for the SC32, we took the approach of manually translating a set of small integer benchmark programs from C to Forth. These programs were collected by the Computer Systems Laboratory at Stanford University and have since been translated from their original Pascal into C. They have been widely used

to evaluate the performance of many computer systems.

Because the Stanford programs are small, they are generally considered "toy" benchmarks that provide overly optimistic results in comparison to similar tests made with larger codes. But several factors suggest that the translated benchmark suite will provide a conservative estimate of performance running large Forth programs.

Merely translating these programs into Forth produced very poor and uncharacteristic Forth code. Word definitions were extremely long and difficult to debug. This meant that the SC32's efficient call/return mechanism was not used. However, our measurements showed that real Forth programs greatly benefit from this feature of the SC32. Array and structure accesses involved run time calculations repeated within inner loops and unnecessary calculations were performed. There are many optimizations traditional compilers perform to minimize this arithmetic. Writing the equivalent program in Forth exposed these excess calculations directly to the programmer. Thus the high level Forth source code would normally be written to avoid these inefficiencies.

Finally, the algorithms and data structures used by the Stanford programs were heavily influenced by traditional languages. A version of one of them, *Towers of Hanoi*, ran 9.6 times faster when coded with data structures and algorithms better suited to Forth than the simple translation of the original code.

The SC32 running with a 10 MHz clock and programmed in Forth was 8.4 times faster on the Stanford benchmarks than a Vax 11/780 programmed in C. If the multiplication dominated *intmm* program is disregarded, then the SC32 is 9.9 times faster. The SC32 is also 19.9 times faster than a 25 MHz Motorola MC68020 running Forth. If the MC68020 is programmed in C than the SC32 is still 1.4 times faster.[3]

Our goal was to develop a processor that could deliver the benefits of an interpreted programming environment without any performance penalty. The data we have collected show that this goal was achieved. Small Forth programs run at least as fast on the SC32 as equivalent C programs on traditional microprocessors. Furthermore it is likely that this relationship will become more favorable for large programs due to the SC32's efficient call/return mechanism.

## 4.5 Applications of the SC32

Several different SC32 based computers have been built at APL. A simple single board computer was designed to demonstrate the chip. That design was later modified and used in telemetry decommutation ground support equipment for the TOPEX and SPINSAT radar altimeter satellites. A standalone computer system, including operating system and utilities, based on magnetic bubble memory for mass storage was developed to show the benefits of self hosted embedded processors for the NASA Goddard Space Flight Center. The most complex SC32 system we have built is a VME bus CPU with full master/slave capability. It will be used to control a balloon borne solar magnetograph. These were interesting projects, but it was not until 1989 that the Freja magnetometer instrument gave us the opportunity to use one of our chips in space flight hardware.

Table 2: Freja Magnetometer Requirements Summary

---

• Anti-alias low pass filters for DC and AC channels
– 64 Hz cutoff during normal rate (14.3 kbits/sec allocated to our instrument) telemetry operations
– 128 Hz cutoff during high rate (28.7 kbits/sec allocated to us) telemetry operations
• Digitize X, Y, Z AC and DC magnetic field measurements to 16 bits
– 128 samples/sec during normal rate telemetry operations
– 256 samples/sec during high rate telemetry operations
• Oversample and average X, Y, and Z DC measurements
• Anti-alias filter one AC channel with 256 Hz cutoff and sample at 512 samples/sec
• Computer amplitude spectrum 0-256 Hz for the AC channel with 512 point FFT
• Detect magnetic activity to trigger data collection in other experiments
• Collect and digitize housekeeping and status data
• Format and output telemetry
• Interpret and execute commands

---

## 5 The Freja Magnetic Field Experiment

Freja is a Swedish satellite that will be launched into a nearly polar orbit to study the earth's magnetosphere and ionosphere. Experiments from Sweden, Germany, and Canada will fly on the satellite and the U.S. is represented by a magnetic field experiment designed and built at APL. Freja is clearly an international effort with launch scheduled in August 1992 as a "piggyback payload" on a People's Republic of China Long March rocket (barring significant changes in the political situation).

### 5.1 Magnetometer Requirements

The magnetometer uses the SC32 to implement the instrument's data acquisition and analysis system. Overall instrument requirements are summarized in Table 2.[4]

The conventional approach to satisfying these requirements would include a switchable hardware anti-aliasing filter (for the two different sample rates), a 16-bit A/D, and an on board computer for status and housekeeping tasks. The processor would be programmed in its assembly language and the code would be cross-assembled on a separate machine. The object code would be downloaded to the target hardware for debugging using in-circuit emulators and other support equipment. No data analysis would be performed on the satellite but would be deferred to ground based postprocessing.

This configuration was not feasible within the resources provided by Freja to our magnetometer. There was neither power nor enough circuit board space for the switchable filters. Filters would also seriously degrade the noise floor of the magnetic field measure-



ments. Telemetry bandwidth precluded transmission of the 512 samples/sec channel to ground for spectral analysis. A separate digital signal processing device used to perform this task would exceed the available power and board space. The extra hardware and software design tasks would also have lengthened our development schedule. Finally, the traditional approach to developing embedded computer software with cross-development tools and in-circuit emulators was too costly due to the long edit, compile, download, and emulate cycle.

Our magnetometer overcame these problems by using a simple fixed hardware anti-aliasing filter, a 16-bit A/D converter, and the SC32 microprocessor. The computer performs data acquisition and averaging, digital anti-alias filtering, FFT computation, telemetry formatting, command interpretation and execution, and other instrument control functions. Software development and debugging were performed interactively on the actual target hardware in a high level language. Despite the processing demands imposed by satisfying these requirements with software, the magnetometer processor has a 50% throughput margin when the SC32 is driven at 40% of its maximum clock rate.

Mass and power requirements were typical of small satellite experiments. The chassis was milled from a solid block of magnesium rather than aluminum and circuit cards were hardwired together instead of using cable assemblies. The completed instrument, excluding probes and boom, weighed 3.5 kg. The entire instrument consumed less than 3.7 W including DC-DC converter, sensor electronics, telemetry subsystem, and the computer itself.

## 5.2 Instrument Development

Schedule and budget constraints were also quite challenging. The flight hardware and software were delivered to Sweden in July 1991, two years after the project was started. We estimate that the hardware and software were developed for 50-75% lower cost than a system of equivalent capability based on a traditional microprocessor such as the 80C86RH. The cost savings were due primarily to our use of an interactive Forth system rather than a cross-compiler/assembler that would be needed for the conventional processor. We also have significant doubt that an equivalent instrument could be based on the 80C86RH due to its limited throughput, even if it were programmed entirely in assembly language.

The productive software development environment provided by the SC32 was a key factor in quickly completing the instrument. Forth's interactive capability greatly assisted hardware debug and subsystem integrations. The flight code was extremely compact, in source (2500 lines) as well as object form (16 Kwords including operating/development system). Small code size was due to two factors. First, our real time scheduler allowed the program to be organized into 8 cooperating tasks. Each task was simple and easily programmed especially when compared to the alternative of a single monolithic piece of code. Secondly, Forth's extensibility meant that program size grew logarithmically as complexity increased. Essentially Forth was used to develop a new programming language specifically oriented to the problem domain. Therefore programs that solved tasks in that domain were very compact. Because of these characteristics of Forth, one of us (Hayes)

### 1.1.10

was able to write the magnetometer flight software in only two months. The magnetometer was delivered in July 1991 and has since been integrated with the other Freja subsystems. We were the first of the seven experiments on Freja to deliver fully flight-ready hardware and software for satellite integration. No flight software changes have yet been needed.

## 5.3 Radiation Testing

Much of the development of our instrument was affected by considerations of the natural radiation environment in Freja's 600 km x 1700 km high inclination orbit. During the two year Freja mission, we expect to receive a total radiation dose of 12 kRad(Si). Radiation induced latch-up and single event upset (SEU) soft errors also concerned us.

The Freja magnetometer CPU board contains the SC32, two 32 K x 32 RAM modules, two 32 K x 32 EEPROM modules, two 82C54RH timer chips, and 52 SSI/MSI parts. The RAM and EEPROM parts were chosen because other APL flight programs had determined that their radiation characteristics were acceptable in Freja's orbit. The 82C54RH radiation tolerance was guaranteed by its manufacturer. SSI/MSI logic from the 54AC00 family were used for the support chips because they were also known, again due to information from other APL flight projects, to work in our environment. We had to establish the radiation characteristics of the SC32 ourselves.

### 5.3.1 Total Dose

Two different SC32 fabrication lots were evaluated for total dose characteristics using our in-house Co<sup>60</sup> facility. Exposure was performed at a rate of 1 kRad(Si)/min with bias and a low speed clock applied to force the part into a known state. Bias current was monitored during exposure. Component functionality was assessed within 1-2 min after each radiation exposure using a standalone computer board executing SC32 diagnostics. Testing required no more than five minutes after each exposure step, thus annealing effects were minimized and the entire test was completed within an hour.

The first lot, obtained from our commercial licensee, was fully functional and within parametric limits beyond 15 kRad(Si) for all five parts tested. The mean total dose tolerance of these parts was 19.9 kRad(Si) with a variance of 4.8 kRad(Si). Full functionality returned overnight to all tested parts from this lot after annealing at room temperature with no bias applied.

The other part lot was supplied directly by our foundry and had been packaged according to Mil-Spec-883B. Our reliability group performed a pre-cap visual inspection of these parts at the foundry and found their quality was excellent and that these parts could easily be upgraded to higher reliability levels through APL's in-house testing and screening procedures. Unfortunately, a process change to improve yield in the two years since the first lot had been built degraded total dose tolerance. Three parts from this lot all failed at slightly more than 5 kRad(Si) when tested with same procedures used with the first lot. An additional three parts were exposed to 1 kRad with two days between subsequent exposures to more nearly simulate the radiation environment of the Freja orbit. These

parts also failed at 5 kRad(Si). Room temperature unbiased annealing has only restored functionality to two of these six parts.

Because of the disappointing total dose behavior of the second batch of parts, we were forced to obtain our flight parts from the first lot. Several factors allowed us to upgrade these commercial parts to space flight quality. The positive report on our foundry's quality control was encouraging, both commercial and Mil-Spec parts were packaged in the same high quality ceramic pin-grid array package, and all lots were assembled with the same equipment and personnel at the foundry. So commercial parts from the first lot were extensively screened at APL and passed all tests.

### 5.3.2 Latch-up and SEU

Radiation induced latch-up and SEU sensitivity of the flight part lot were also evaluated. Initially, SC32 parts were screened for latch-up sensitivity in an in-house  $\text{Cf}^{252}$  chamber. This equipment exposed the die to heavy ions with a mean linear energy transfer (LET) of 36 Mev-cm<sup>2</sup>/mg at a high flux rate. The SC32 did not latch during a 30 minute exposure. Subsequent work showed that many other chip types also did not latch in the  $\text{Cf}^{252}$  chamber.

However, later tests made at the Single Event Upset Test Facility of the Brookhaven National Laboratory Tandem Van de Graaff accelerator cast doubt on conclusions about latch-up sensitivity based on  $\text{Cf}^{252}$  data. Using the Brookhaven equipment we were able to gather both radiation induced SEU and latch-up sensitivity of the SC32. The chip did latch-up with an LET threshold of 15.6 Mev-cm<sup>2</sup>/mg which corresponds to about 1 latch-up per 21 years in the Freja orbit. An SEU threshold of 5 Mev-cm<sup>2</sup>/mg was also observed which was estimated to be equivalent to one soft error every 166 days in our orbit.

These radiation testing results led us to add latch-up protection circuitry to the DC-DC converter. If excessive current is drawn by the SC32, the CPU board will be momentarily turned off thus resetting the latched circuitry. After power is restored the computer will resume normal processing.

SEU events are more difficult to detect and their impact can be more subtle. An SEU could disturb the program controlling the processor or it could invalidate a single word of science data. Because an SEU is only expected every few months, it represents only a minor error in the collected data and will be ignored. Program errors will be detected by a watchdog timer that must periodically be updated. An SEU induced program error will most likely be detected by a failure to properly access the watchdog. In response, the watchdog will reboot the system. Both types of radiation induced error should occur rarely enough that these correction strategies will not significantly degrade the quality of the magnetometer data.

## 6 Conclusions

Because the SC32 was originally designed as a research effort and was only manufactured by a commercial foundry, many questions had to be resolved before we could use it a space

based instrument. Reliability concerns were greatly reduced after a site visit to the foundry showed excellent manufacturing procedures were followed. A thorough screening of parts from the flight lot has also added to our confidence in the reliability of the SC32.

Radiation tolerance of our chip was also studied. Early testing of our prototype chips indicated they would meet our needs. Commercial versions of our chip manufactured shortly thereafter were fully evaluated and had acceptable radiation tolerance. However, the foundry modified the manufacturing process to improve yield in the interval between when our prototypes were evaluated and when we ordered Mil-Spec chips for our instrument. This process change had the unfortunate side effect of diminishing total dose tolerance to unacceptable levels. Unless a foundry rigorously controls those aspects of the process that impact radiation tolerance, performance may vary significantly between lots.

We have shown that a Forth language directed microprocessor with hardware and software optimized for embedded systems can significantly improve spacecraft instrumentation. Because of the capabilities of the magnetometer's computer based on the SC32, an instrument of unprecedented capability was developed at far lower cost than could otherwise be achieved.

The most important lesson we have learned from this work is that a custom integrated circuit of the right architecture can deliver substantial benefits even when only one chip is needed. System performance that is unreachable with catalog components can be achieved and qualification issues can be resolved. Most surprisingly, system development costs can be reduced by using custom chips. Savings from designing fewer circuit boards, consuming less power, buying fewer expensive flight components, and most importantly greater software productivity easily balance the additional costs of developing and qualifying the right custom integrated circuit.

## 7 Acknowledgements

In addition to the authors, Susan Lee, Susan Waters, Mary Wong, and Tom Zaremba have all contributed to this work over the past decade. We greatly appreciate the advice and assistance received from many people in APL's SOR group in making our instruments as reliable as possible. The many talents our shop groups contributed to this work were also vital to our success. Finally, we would like to thank Larry Zanetti, the Freja magnetometer's principal investigator, for encouraging us to use the SC32 in his instrument, and our management, particularly Jay Dettmer and Tom Zaremba, for their support and encouragement of our efforts.

## References

- [1] B. Ballard and J. Hayes, Forth and Space at the Applied Physics Laboratory, *Proc. of the 1991 Rochester Forth Conference*, Inst. of App. Forth, Rochester, N.Y., June 1991.

- [2] J. Hayes and S. Lee, The Architecture of the SC32 Forth Engine, *J. of Forth App. and Res.*, V5, N4, pp. 493-506.
- [3] M. Fraeman, Performance Evaluation of the SC32 Stack Microprocessor, *Proc. of the 1989 Rochester Forth Conference*, Inst. of App. Forth, Rochester, N.Y., June 1989.
- [4] R. Henshaw, B. Ballard, J. Hayes, and D. A. Lohr, An Innovative On-Board Processor for Lightsats, *Proc. of the 4<sup>th</sup> AIAA/USU Conf. on Small Satellites*, AIAA, August 1990.



## **Multi-chip Modules: A High-performance Packaging Alternative**

L. Salmon  
Brigham Young University

*Abstract-* Multi-chip Module (MCM) packaging has emerged as an important technology for high-performance electronic systems. Benefits of MCMs include: high IC packing density, low interconnect propagation delay, excellent power dissipation characteristics, and low cost. This paper will review MCM substrate fabrication, testing, and design. Major challenges for MCM implementation in high-performance systems will be discussed. Finally, applications of MCM technology to current high-end computer systems will be reviewed.

